

[0147] The method 400 begins at step 405 where the processor 205 constructs an ACK_DOC message from predetermined ACK_DOC opcode. Any suitable opcode for implementing the ACK_DOC message and the other messages described herein can be used. Such opcode will not be described in detail in this document. The ACK_DOC message is constructed using the version information embedded within the document (i.e., the username of the author, the username of the acknowledger, the UUID, the current version string, and optionally, a document hash resulting from applying any suitable hash algorithm (e.g., MD5 or SHA) to the document). At the next step 410, the ACK_DOC message is transmitted over the network 220 to all of the collaborators listed in the list of active collaborators, configured within memory 206, associated with the document. A method 1600 of broadcasting the ACK_DOC message, as executed at step 410, will be described below with reference to FIG. 16. Following step 410, execution returns to step 101 of the method 100.

[0148] The method 500 of modifying the version information embedded within the document, as executed at steps 310 and 338 of the method 300, will now be described with reference to the flow diagram of FIG. 5. The method 500 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0149] The method 500 includes only one step 505. At step 505, the username of the collaborator who acknowledged receiving the document, (i.e., the originator of the ACK_DOC message) is added to the list of acknowledgers for the current version of the document, which is embedded within the document. Following step 505, execution returns to step 312 or 340 of the method 300, depending on whether the method 500 was invoked from step 310 or step 338 of the method 300.

[0150] The method 600 of transmitting an ACK_LOGIN message, as executed at step 318, will now be described with reference to FIG. 6. The method 600 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0151] The method 600 begins at step 605, where the processor 205 constructs the ACK_LOGIN message from a predetermined opcode. The ACK_LOGIN message includes the username of the collaborator acknowledging the login and the UUID of the document to which the collaborator has logged in. The method 600 then proceeds to step 610 where the ACK_LOGIN message is transmitted to the originator of the LOGIN message. Following step 610, execution returns to step 101 of the method 100.

[0152] The method 700 of broadcasting an ACK_MOD message over the network 220 to all active collaborators, as executed at step 320, will be now described below with reference to FIG. 7. The method 700 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0153] The method 700 begins at step 705, where the processor 205 constructs the ACK_MOD message. The ACK_MOD message includes the embedded version information associated with the document including the username of the author of the document, the username of the collaborator acknowledging the modification, the document

UUID, the current version string, and, optionally, a modification signature for verification purposes. The modification signature can be configured using any known method of generating a digital signature or hash. At the next step 710, the processor 205 broadcasts the message to all active collaborators over the network 220, in accordance with a method 1600 to be described below with reference to FIG. 16.

[0154] The method 800 of transmitting a REQ_MOD message to request a patch from collaborators, as executed at step 334, will be described in detail below with reference to FIG. 8. The method 800 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0155] The method 800 begins at the first step 805, where the REQ_MOD message is constructed from predetermined opcode. The REQ_MOD message includes the username of the originator of the REQ_MOD message (i.e., the requestor), the UUID of the document, and the version string of the version of the document which is currently stored in memory 206 and which needs to be updated. Execution then returns to step 101 of the method 100.

[0156] A method 900 of determining the difference (i.e., a patch) between the latest version of the document stored in memory 206 (i.e., the document loaded from the hard disk drive 210) and the version of the document that has been identified in the REQ_MOD message, as executed at step 344 (and step 1315 of FIG. 13), will now be described in detail below with reference to FIG. 9. The method 900 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0157] The method 900 begins at step 905, if the processor 205 determines that there is a component (i.e., a current component) of the document stored in memory 206 that is different from a corresponding component in the latest version of the document and requires determination, then the method 900 proceeds to step 910. Otherwise, execution returns to step 346 of the method 300 (or to step 1320 of the method 1300), depending on whether the method 900 was invoked from step 344 (or step 1315). At step 910, a delta value for the current component is set to the difference between the value for the current component for the document stored in memory 206 and the version of the document identified by the REQ_MOD message. Execution then returns to step 905. A patch determined can include one or more delta values, determined in accordance with the method 900, corresponding to one or more document components.

[0158] The method 1000 of transmitting a MODIFY message, as executed at step 346, will now be described with reference FIG. 10. The method 1000 is preferably implemented as software resident on the hard disk drive 210 and being controlled in its execution by the processor 205.

[0159] The method 1000 begins at step 1005, where the MODIFY message is constructed. The MODIFY message includes the patch determined at step 344, the username of the collaborator who created the patch, the UUID for the document, a version string corresponding to the version of the document to which the update applies, and a further version string corresponding to the version to which the